INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Maneuver-Based Obstacle-Avoiding Trajectory Optimization

Dirk von WISSEL
Ramine NIKOUKHAH - François DELEBECQUE

N° 1955
Juillet 1993

PROGRAMME 5

Traitement du signal,
automatique et
productique

Rapport de recherche

1993

Maneuver-based obstacle-avoiding trajectory optimization

Dirk von Wissel, Ramine Nikoukhah and François Delebecque

INRIA
ROCQUENCOURT BP 105
78153 LE CHESNAY CEDEX, FRANCE

ABSTRACT

The problem of maneuvering complex vehicles through obstacles is considered. The proposed solution consists in searching the optimal path through a tree of trajectory segments. Since these segments of trajectory, called maneuvers are precalculated this method can be used for real-time applications very efficiently. The approach is illustrated through the example of a riderless bicycle.

Optimisation de trajectoires: contournement d'obstacles à partir de manoeuvres

RESUME

On considère le problème du guidage d'un véhicule à travers des obstacles. La solution proposée est de construire un arbre de recherche à partir de segments de trajectoires (manoeuvres). Comme les manoeuvres sont précalculées la méthode peut être utilisée d'une manière très efficace pour des applications en temps réel. On présente la méthode sur un modèle de vélo autoguidé.

1 Introduction

In this paper, we consider the problem of automatic real-time obstacle-avoidance and trajectory optimization for a complex vehicle equipped with sensing devices, such as cameras, radars, sonars ,etc..., which allows it to learn continuously about its environment (possible obstacles) ahead, up to a given distance. Our objective is to design a controller for this vehicle which would allow it to maneuver itself through obstacles while moving in a given general direction or towards a given point. As a second objective the controller is to choose among all admissible trajectories the one that minimizes some cost function such as time, distance, etc...

The complexity of the vehicle's dynamics is supposed to be such that it forbids any approach that requires real-time manipulation of its dynamic model. But this is exactly what is needed for constructing the optimal solution to our problem because, unlike most path planning and trajectory optimization problems considered in the literature (see for example [1]), due to short-sided vision of our vehicle, the controller has to update the trajectory in real-time.

The standard approach to constructing sub-optimal solutions to any such problem would be to decompose the problem into two problems. The first problem consists in the construction of a candidate path that avoids all the obstacles, and the second, construction of a feedback controller which tries to keep the vehicle as close as possible to this path (to track the path)(see for instance [2],[3],[4],[5]). The problem with this approach is that no matter how well the feedback controller is designed, there is no guarantee that the vehicle would stay close enough to the desired path (in particular, we could not set an upper bound on the distance that separates the desired path and the actual path). We can make the candidate path more trackable by taking into account the dynamics of the vehicle in more details and thus increasing computational complexity. However it remains the difficulty that we cannot guarantee obstacle avoidance unless complete dynamics are taken into account in the construction of the candidate path ,something which is not possible in our framework.

Here we propose a slightly different approach for constructing the controller. We still break the problem up in two, but we make sure that in the first step, the path which is generated is perfectly trackable. For that we start by constructing a set of maneuvers. Maneuvers are trajectory segments which have the property that they can be connected in any order giving a perfectly trackable trajectory. For example for the case of the bicycle which we shall consider in this paper, maneuvers are pieces of trajectory that begin in the straight up position (no lean, no lean velocity) with straight ahead handlebars and end in the same position.

Trajectories that can be constructed from maneuvers form a tree. In this tree we pick the trajectory (the path) that avoids all obstacles and minimizes the cost and use it as the candidate trajectory. Since the maneuvers are computed off-line, open-loop controls that execute them can be stored in memory and thus reducing considerably the complexity of the controller. Efficient algorithms for constructing the set of maneuvers picking the right path in the tree and updating the path based on new information in real-time will be presented.

The maneuver - based trajectory optimization is clearly suboptimal because we only consider trajectories that are obtained as a sequence of maneuvers. The richer we choose the set of maneuvers, the closer we shall be to optimality.

The outline of the paper is as follows. In Section 2 we study conditions under which our two stage approach can work; in particular, we formally define notions of separability and maneuver stability which are essential elements in our approach. The major steps in the design of the controller are presented in Section 3. In Section 4 we illustrate our approach by considering thoroughly the example of the bicycle.

2 Separability and maneuver stability

In order to be able to break up the optimization problem at hand into two subproblems, as we have proposed, we need to have certain properties. In particular what we need is that the space over which the vehicle is traveling be homogeneous (flat), except of course for the obstacles. In that case the vehicle behaves the same way to the same commands (turning, acceleration, leaning, etc...) regardless of its location and orientation provided it has the same initial configuration[1] (for example in the case of the bicycle: the configuration corresponds to its lean angle, handlebars' angle with respect to the frame, the associated velocities and the speed; in the case of a truck with a trailer: the angle between the truck and the trailer, the position of the steering wheel, the associated velocities and the speed of the vehicle). When this condition is satisfied, we say that the system is separable. In the separable case, it is very natural to consider constructing trajectories by concatenation of segments of trajectory (maneuvers). These maneuvers, however, depend on the initial configuration so that to be able to piece together different maneuvers, we should make sure that maneuvers start and end in a given configuration which we shall call neutral (a generalization to the case of non-unique but finite neutral configurations is possible). Thus the set of maneuvers consists of segments of trajectories and the associated set of commands that allows the vehicle in neutral configuration to follow the segment and be at the end of the segment in neutral configuration. For example for the case of the bicycle having for control just a torque on the handlebars the neutral configuration could be a straight-up position, zero lean-velocity with handlebars perpendicular to the frame with zero turning velocity and a given speed (in the neutral configuration, control set to zero, the bicycle would follow a straight path). Maneuvers are then commands that turn the bicycle such that at the end of the maneuvers we are in neutral configuration. This is essentially what a rider would do on a bicycle.

Even though, theoretically, we can construct trajectories out of maneuvers, in practice, the requirement that at the start of each maneuver the vehicle must be in neutral configuration adds a new requirement namely that the maneuvers be stable, which means that if the vehicle is not exactly in neutral configuration but slightly perturbated at the start of the maneuvers, at the end, the vehicle must be closer to (at least not further from) the neutral

[1] Generalized coordinates and velocities of the vehicle which are independent of its absolute position and orientation on the plane.

configuration. This stability condition guarantees that a slight perturbation does not have catastrophic effects. If maneuvers are not stable, either we should look for another set of maneuvers or add a feedback control stabilizing the neutral configuration to the vehicle (in which case its effects in the behavior of the vehicle must be taken into account) and then construct the set of maneuvers. For the example of the bicycle, below certain speed, even the simplest and most trivial maneuver which consists of applying no torque (and thus going straight) is not stable. In this case, we can construct a feedback controller and add it to the unstable bicycle and then construct the set of maneuvers for this new system (of course stabilizing just the straight line maneuver alone does not guarantee that the other maneuvers are also stable; it must be verified). We shall see how maneuver stability can be tested when we consider the bicycle example.

3 Controller structure

The first sep in the design of the controller is the selection of maneuvers. Maneuvers can either be computed through analytical methods, or optimization algorithms (as we do in the case of a bicycle); it can even be constructed by observing a human operator. If needed, a feedback maneuver stabilizer should be included.

The second step is a tree search problem. The set of trajectories constructed out of maneuvers is clearly a tree. A tree which depending on the number of maneuvers, can grow very rapidly so that it would not be possible to do a complete search for picking the optimal path. We can however use the idea of Dynamic Programming to reduce the number of branches. We start by discretizing the position space (which consists usually of the x-y-coordinates and the orientation of the vehicle) into what we shall call cells. Then if two subtrajectories end in the same cell, we compute their cost and throw away the one which has a higher cost. The actual organization of the algorithm, which is an A^* (see for example [6]) algorithm, is given in the Appendix A.

Finally, it would obviously be unreasonable to assume that once we have planned a trajectory, by applying the corresponding controls, the vehicle would follow it exactly even if the trajectory is perfectly trackable. Just consider a small perturbation in the initial state. Such a perturbation clearly affects the actual trajectory and can cause an error which can grow linearly with time (speed of the vehicle being constant). For this reason but also due to the short-sighted vision nature of the system, the controller strategy must be of a sliding window type. That means that the vehicle observes the obstacles ahead up to a distance and plans a trajectory. But, it does not follow it all the way, rather it performs a few maneuvers and makes a new observation (positions itself) then replans a new trajectory an so on. How often the controller updates the trajectory has to do with computational power available.

To summarize, the structure of the controller is as follows: first, if needed, a continuous feedback stabilizing controller is implemented to guarantee maneuver-stability. Then a hybrid open-loop-closed-loop controller is used to maneuver the vehicle through obstacles.

4 Example of a riderless bicycle

We will illustrate our approach in this section through an academic but concrete example: control of a riderless bicycle. The methods used can easily be extended to any vehicle with a dynamic model that can be described by an algebraic-differential system (thus any vehicle made of rigid parts).

number of generalized coordinates: 23

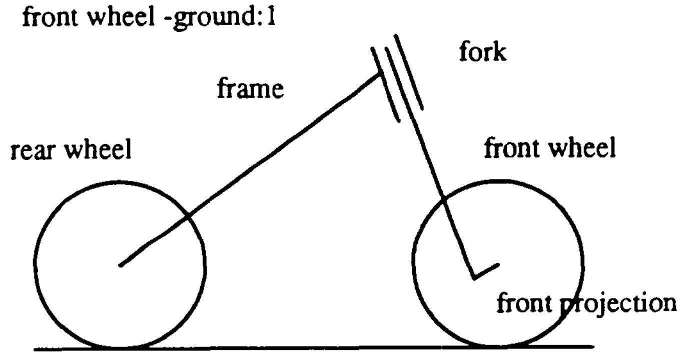
control: torque on fork (handel-bars)

holonomic constraints:

rear wheel -frame : 3
frame - fork: 5
fork-front projection:3
front projection -front wheel :3
rear wheel-ground : 1
front wheel -ground:1

non-holonomic constraints:

rear wheel-ground: 2
front wheel-ground: 2



Locally 3 degrees of freedom (23-(3+5+3+3+1+1+2+2)):
move back and forth; lean and turn the fork (handlebars).

Globally 7 degrees of freedon: (23-(3+5+3+3+1+1)):
reachable space

Figure 4.1 *Simplified model of a bicycle.*

4.1 Model of a bicycle

The dynamics of any rigid multibody mechanical system can be described by an implicit non-linear system

$$\mathbf{E}(\xi)\dot{\xi} = \mathbf{F}(\xi, u) \tag{4.1}$$

where ξ describes the generalized coordinates, velocities and internal forces associated with the holonomic and nonholonomic constraints (Lagrange multipliers), and, u, the external forces and torques (control inputs). The model of the bicycle that we use [7] is obtained using the Lagrangian formalism (see Appendix B).

For simplicity, we assume that the system has only one input which is a torque on the handlebars.

4.2 Maneuver construction

A maneuver in this case corresponds to a control which makes the bicycle (in its neutral configuration) turn a given angle (and end up in a neutral configuration). If in addition, we require that the maneuvers be as short as possible (in time), we end up with an optimization problem which has a bang-bang solution. To solve this optimization problem, the idea of our algorithm is to fix the number of switchings between the minimum and maximum-value of the input function and to optimize just the switching-times. Since by applying standard index reduction techniques, i.e. by differentiating a sufficient number of times the constraints (see for example [7]), the implicit representation (4.1) can be transformed into an explicit description, we shall suppose from here on that the model of system is given by:

$$\dot{\xi} = f(\xi, u) \quad , \quad \xi(0) = \xi^0 \quad . \tag{4.2}$$

Among the input-functions with $k-1$ commutations

$$u(t) = \begin{cases} u_I & \text{for} \quad 0 \le t \le t_1, \quad t_2 \le t \le t_3, \quad \cdots \quad t_{k-2} \le t \le t_{k-1} \\ u_{II} & \text{for} \quad t_1 \le t \le t_2, \quad t_3 \le t \le t_4, \quad \cdots \quad t_{k-1} \le t \le t_k \end{cases} , \tag{4.3}$$

$$t_j = \sum_{i=1}^{j} \tau_i$$

where u_I and u_{II} represent respectively the upper and lower bounds on the control function, we search the optimal parameter-vector $\tau = [\tau_1, \cdots, \tau_k]$ minimizing the cost-criterion

$$J = \sum_{i=0}^{k-1} \int_{t_i}^{t_{i+1}} g_i(\xi(t))dt + h(\xi(t_k)) \, . \tag{4.4}$$

subjected to constraint (4.2). A variation δJ yields

$$\begin{aligned} \delta J &= g_1(\xi(t_1))\delta\tau_1 - g_2(\xi(t_1))\delta\tau_1 - \\ &- g_2(\xi(t_2))\delta t_2 + g_3(\xi(t_3))\delta t_3 - \\ &\vdots && (4.5) \\ &+ \frac{\partial h}{\partial \xi}(\xi(t_k))\delta\xi(t_k) && (4.6) \end{aligned}$$

When $k = 1$ we get

$$\delta J = g_1(\xi(t_1))\delta\tau_1 + \frac{\partial h}{\partial \xi}(\xi(t_1))\delta\xi(t_1) \ . \tag{4.7}$$

With $\delta\xi(t_1) = f(\xi(t_1), u_I)\delta\tau_1$ which results immediately from (4.2), the gradient of J with respect to τ_1 becomes

$$\frac{\partial J}{\partial \tau_1} = g(\xi(t_1)) + \frac{\partial h}{\partial \xi}(\xi(t_1))f(\xi(t_1)). \tag{4.8}$$

By the application of standard gradient-optimization methods, the optimal value for τ_1 can be computed ($\xi(t_1)$ is obtained by integrating the system).

The case $k = 2$ requires some additional considerations. We get from (4.6) that

$$\delta J = g_1(\xi(t_1))\delta\tau_1 - g_2(\xi(t_1))\delta\tau_1 + g_2(\xi(t_2))\delta(t_2) + \frac{\partial h}{\partial \xi}(\xi(t_2))\delta\xi(t_2) \tag{4.9}$$

with $\delta\xi(t_2) = \delta_{\tau_1}\xi(t_2) + \delta_{\tau_2}\xi(t_2)$. The part $\delta_{\tau_1}\xi(t_2)$ represents the variation of ξ at $t_2 = \tau_1 + \tau_2$ induced by the variation of the initial-condition at $t_1 = \tau_1$ (which itselfs results from a variation over the first interval τ_1). The effect of the variation of τ_1 at t_2 is determined by the linear relation $\delta_{\tau_1}\xi(t_2) = \mathbf{Z}_1\delta\xi(t_1)$, where matrix $\mathbf{Z}_1$ is the propagation-matrix of the perturbation which is determined by the solution of the linearization of (4.2) over the τ_2 interval.

$$\dot{\mathbf{Z}}(t) = \frac{\partial f}{\partial \xi}(\xi(t), u_I)\mathbf{Z}(t) \ , \ \mathbf{Z}(t_1) = \mathbf{I} \tag{4.10}$$

$$\mathbf{Z}_1 = \mathbf{Z}(t_2) \ .$$

Using the relations $\delta\xi(t_1) = f(\xi(t_1))\delta\tau_1$ and $\delta\xi(t_2) = f(\xi(t_2))\delta\tau_2$ as in the case $k = 1$, reporting $\delta_{\tau_2}\xi(t_2) = f(\xi(t_2), u_{II})\delta\tau_2$ and $\delta_{\tau_1}\xi(t_2) = \mathbf{Z}_1 f(\xi(t_1), u_I)\delta\tau_1$ in $\delta\xi(t_2) = \delta_{\tau_1}\xi(t_2) + \delta_{\tau_2}\xi(t_2)$ and using (4.9) we get the gradient of J with respect to τ_1 and τ_2.

In the general case the variation of ξ at $t = t_k$ is

$$\delta\xi(t_k) = \sum_{i=1}^{k} \delta_{\tau_i}\xi(t_k) \tag{4.11}$$

$$\delta_{\tau_i}\xi(t_k) = \left(\prod_{j=1}^{k} \mathbf{Z}_j\right) f(\xi(t_i), u(t_i))$$

$$\mathbf{Z}_j = \mathbf{Z}(t_{j+1}) \ , \ \text{solution of } \ \dot{\mathbf{Z}}(t) = \frac{\partial f}{\partial \xi}(\xi(t), u(t))\mathbf{Z}(t) \ , \ \mathbf{Z}(t_j) = \mathbf{I} \ , \text{ for } j = 1, \cdots, k-1 \ ,$$

$$\text{for } j = k \text{ we have } \mathbf{Z}_k = \mathbf{I} \ ,$$

which reported in (4.6) determines the gradient with respect to the parameter-vector $\tau = [\tau_1, \cdots, \tau_k]$.

For the bicycle with a speed of $v = 8\text{m/s}$ in neutral configuration and the two values $u_I = -0.5\text{Nm}$, $u_{II} = 0.5\text{Nm}$ for the lower and upper bounds on the torque on the handlebars we have chosen the following set of maneuvers (the unit of all intervals τ_i is one second):

- 45 degrees to the left or right
  - For the parameter vector τ of optimal switching-time we obtain:
    $\tau = [2.7700514, 0.1544306, 0.2173454, 0.1720983, 0.0534789, 0.0116612]$
- 15 degree to the left or right
  - To show that even maneuvers can be constructed from other maneuvers here the maneuver itself is composed of three identical maneuvers of 5 degree to the left: For the maneuver of 5 degree to the left the parameter vector τ of optimal switching-time is:
    $\tau = [0.3596196, 0.1489125, 0.2417108, 0.1322752, 0.0126554]$
- straight line,"keep neutral configuration"
  - The neutral configuration is kept during the intervall $\tau = 2.6500000$.

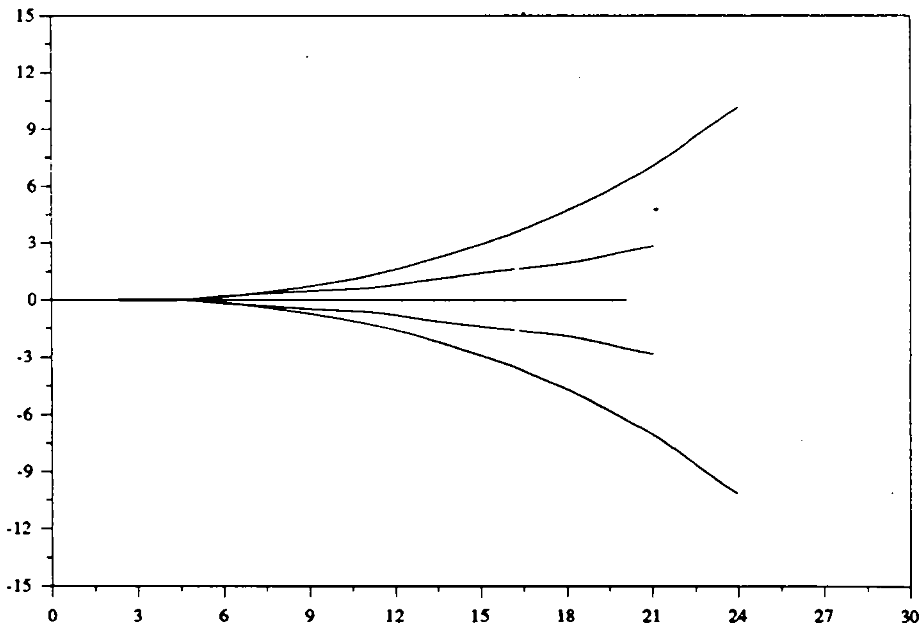


Figure 4.2 *Set of maneuvers projected on the x-y plane.*

In Figures 4.3 and 4.4 the behavior of the bicycle during the execution of the first two maneuvers is illustrated.

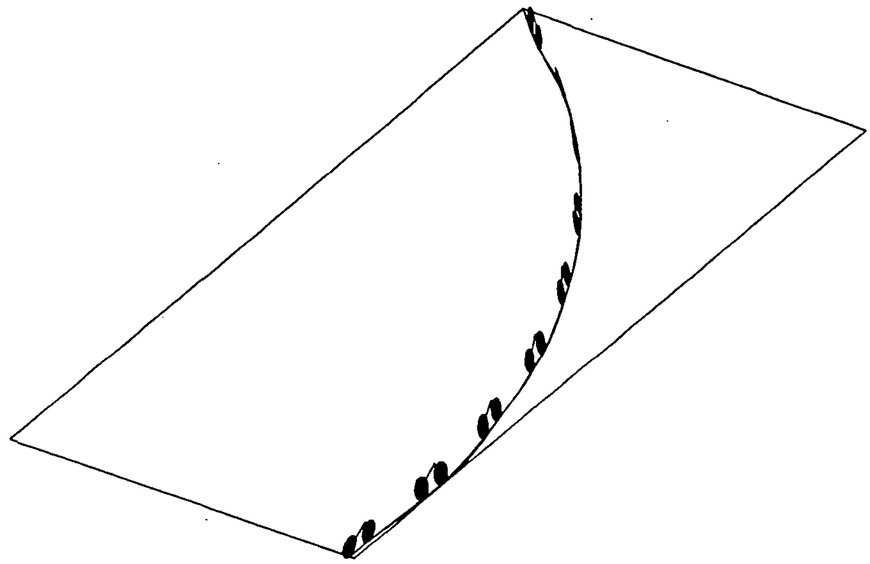


Figure 4.3 *Maneuver turning 45 degree to the left.*

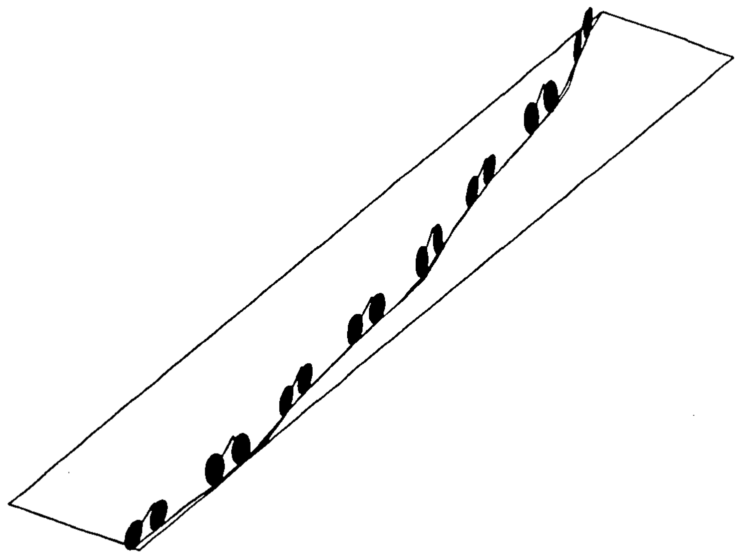


Figure 4.4 *Maneuver turning 15 degree to the left. The maneuver itself is a composition of tree maneuvers of 5 degrees to the left.*

4.3 Maneuver stability

As we have seen, a maneuver goes from a neutral configuration to a neutral configuration. Let us denote bicycle's configuration by χ (which is clearly a projection of ξ),

$$\chi(t) = P(\xi(t)) \tag{4.12}$$

for some function P. If T_i denotes maneuver i's duration, we must have, after the execution of maneuver i

$$\chi(T_i) = \chi_0 = \chi(0) \quad , \tag{4.13}$$

where χ_0 denotes the neutral configuration. For our approach to work, we must have that a perturbation in the configuration does not propagate through the application of successive maneuvers. Let the function Ψ_i be such that

$$\chi(T_i) = \Psi_i(\chi(0)) \quad , \tag{4.14}$$

where $\chi(T_i)$ denotes the configuration of the bicycle at the end of maneuver i. Note that due to the separability condition, the dynamics of the configuration of the bicycle is independent of its positions and orientation; i.e, the configuration has its own dynamics

$$\dot{\chi}(t) = \psi(\chi(t), u(t)) \quad . \tag{4.15}$$

Clearly

$$\Psi_i(\chi_0) = \chi_0 \tag{4.16}$$

for all i (maneuvers), so that if the system is not initially in neutral configuration χ_0 but slightly off

$$\chi(0) = \chi_0 + \delta\chi \tag{4.17}$$

then its configuration after the execution of the maneuver i would be

$$\chi(T_i) = \chi_0 + \frac{\partial \Psi_i}{\partial \chi}(\chi_0)\delta\chi + \sigma(\delta^2) \quad . \tag{4.18}$$

This means that, after successive execution of maneuvers $i_1, i_2, \cdots, i_N$, the perturbation at the end would be

$$\delta\chi(\sum_{j=1}^{N} T_{i_j}) = \prod_{j=1}^{N} \frac{\partial \Psi_i}{\partial \chi}(\chi_0)\delta\chi \quad . \tag{4.19}$$

Clearly a sufficient condition for maneuver stability is that

$$\overline{\sigma}(\mathbf{\Delta}_i) < 1 \tag{4.20}$$

for all maneuvers i, where

$$\mathbf{\Delta}_i = \frac{\partial \Psi_i}{\partial \chi}(\chi_0) \quad . \tag{4.21}$$

To compute $\mathbf{\Delta}_i$ note that if Q is any function such that $P(Q)$ is the indentity function, then

$$\psi(\chi) = \frac{\partial P}{\partial \xi} f(Q(\chi, u) \tag{4.22}$$

where f is as defined in (4.2). Using function ψ, we can compute $\mathbf{\Delta}_i$ by noting that

$$\delta\dot{\chi} = \frac{\partial\psi}{\partial\chi}(\chi, u_i)\delta\chi \ , \tag{4.23}$$

where $u_i(t)\,,\, 0 < t < T_i$, denotes the control associated with maneuver i. This implies that

$$\delta\chi(t) = \mathbf{\Phi}_i(t)\delta\chi(0) \ , \tag{4.24}$$

where $\mathbf{\Phi}_i$ is given by

$$\dot{\mathbf{\Phi}}_i(t) = \frac{\partial\psi}{\partial\chi}(\chi(t), u(t))\mathbf{\Phi}_i(t) \quad , \quad \mathbf{\Phi}(0) = \mathbf{I} \ . \tag{4.25}$$

It is then easy to see that

$$\mathbf{\Delta}_i = \mathbf{\Phi}(T_i) \ . \tag{4.26}$$

Thus, to test stability of maneuver i, we should integrate (4.15) with $u = u_i$ and (4.25); use (4.26) to compute $\mathbf{\Delta}_i$ and verify (4.20) for every i. If (4.20) fails for any i, we should either discount maneuver i from the set of maneuvers or go back and add a stabilizing feedback to the bicycle.

For the chosen set of maneuvers we obtain the following maximal singular values of the propagation matrixs Δ_i:

- 45 degree to the left or right
  - corresponding maximal singular value: $\max \overline{\sigma}(\mathbf{\Delta}_1) = 0.575$
- 15 degree to the left or right
  - corresponding maximal singular values $\max \overline{\sigma}(\mathbf{\Delta}_2) = 0.605$
- straight line, "keep neutral configuration"
  - corresponding maximal singular values: $\max \overline{\sigma}(\mathbf{\Delta}_3) = 0.578$

5 Simulation results

Figure 5.1 shows the first optimization problem solved by the A^* algorithm. The grey area shows the end of the sliding window.

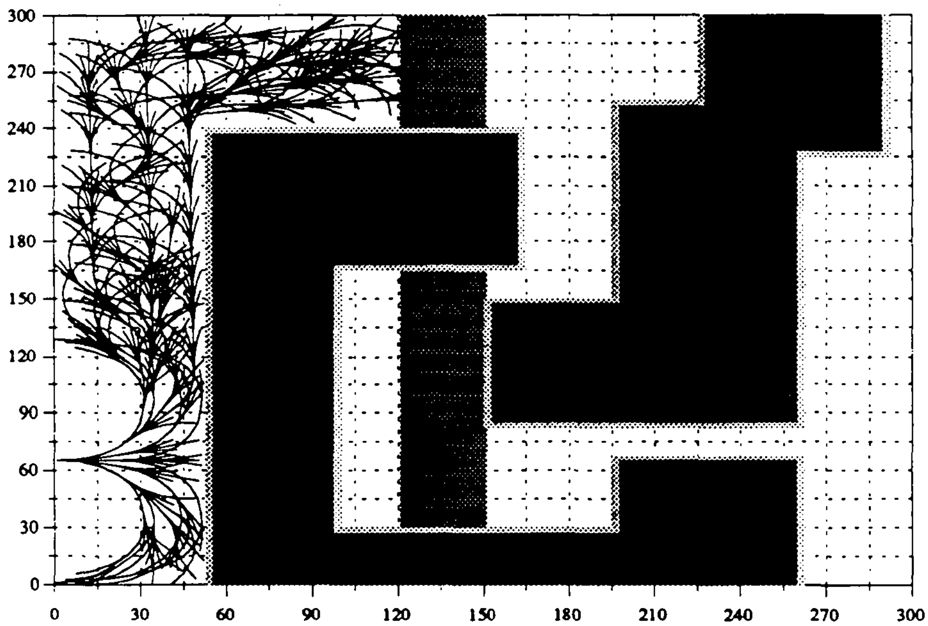


Figure 5.1 *Initial point is* $A = A' = (x = 0, y = 65, \phi = 0)$.

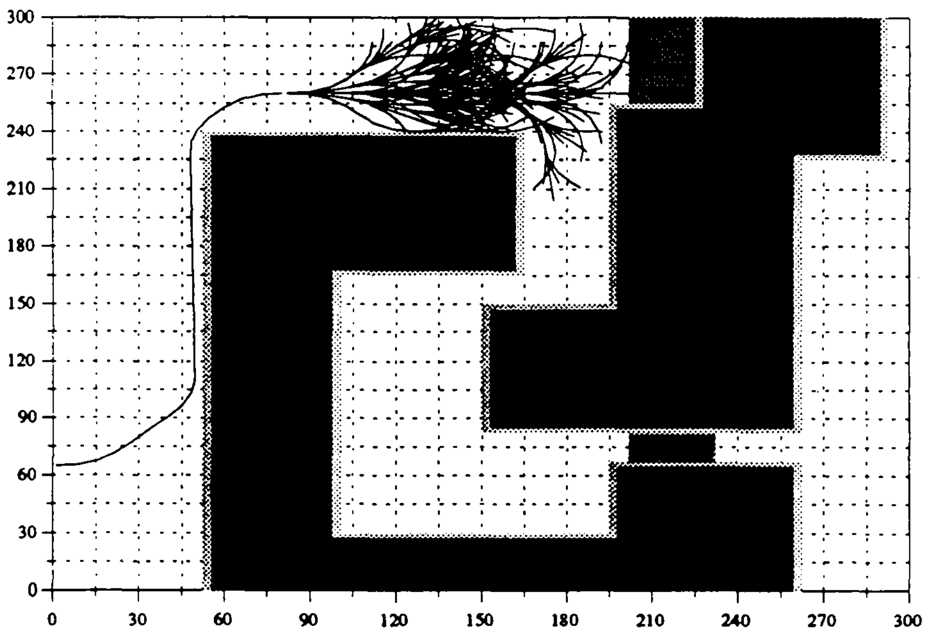


Figure 5.2 *Initial-point is* $A' = (x = 241, y = 159, \phi = -\pi/4)$. *The trajectory* $A\ A'$ *is the already covered path.*

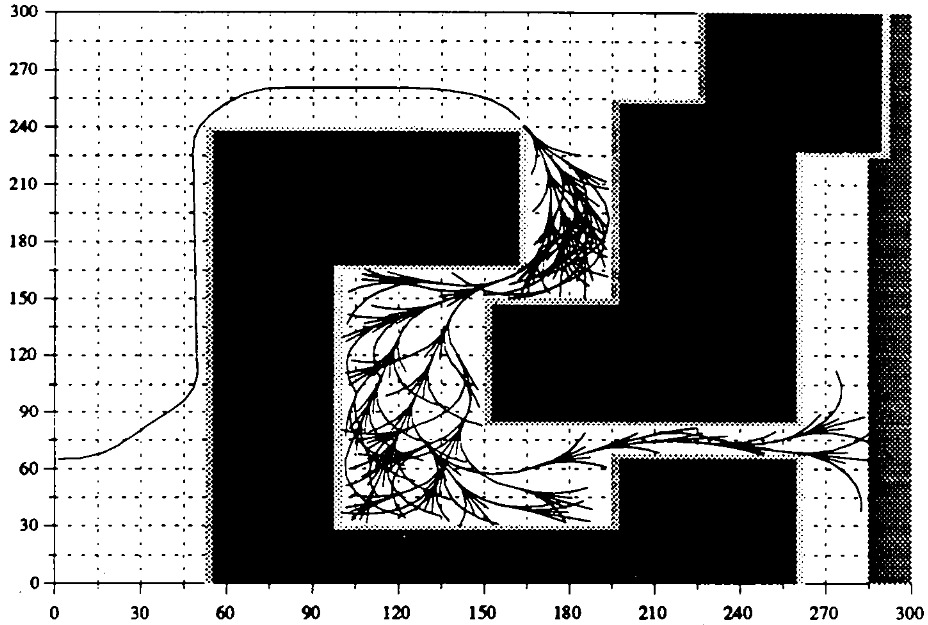


Figure 5.3 *Tree after a number of iterations. The initial-point is* $A' = (x = 110, y = 248, \phi = 0)$. *The trajectory A A' is the already covered path.*

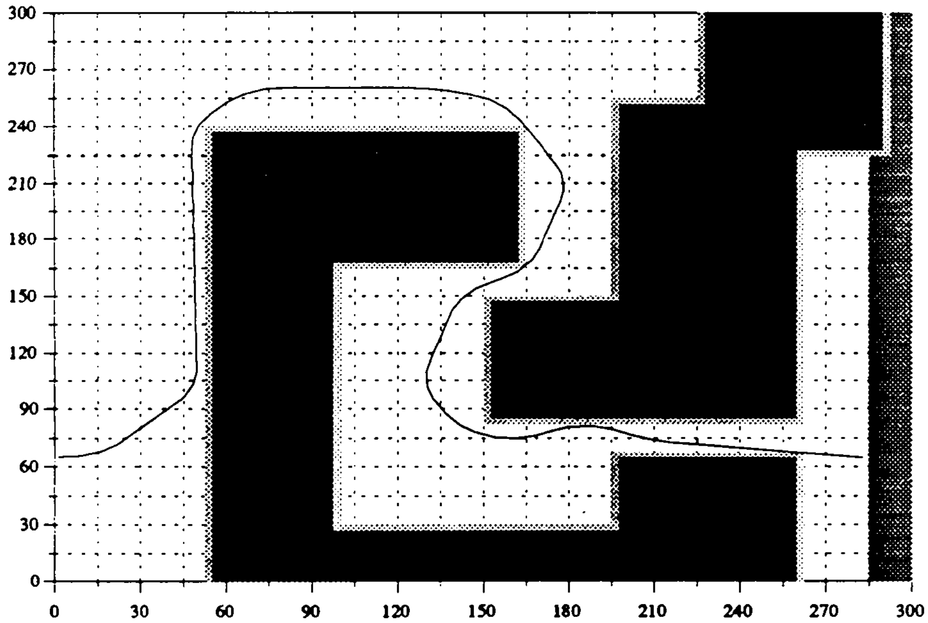


Figure 5.4 *Path found by the maneuver-based obstacle avoiding pathplanning algorithm.*

6 Conclusion

In this paper, we have presented a methodology for constructing controllers for maneuvering a vehicles through obstacles while moving in a given general direction. We have illustrated the approach through the example of a bicycle, launched with an initial velocity, where for only control we have an actuator placed at the joint between the frame and the fork (so that the input is a torque on the handlebars). This example has been chosen because, despite its simplicity (only four rigid bodies and one input) it illustrates perfectly all the steps necessary for the construction of the controller.

Of course for more realistic applications, we have in general more than one input and very often more than one neutral configurations (corresponding for example to various speeds). having more inputs and neutral configurations implies necessity for a richer set of maneuvers. Moreover, when we have more than one neutral configuration, we should make sure that we only consider trajectories that are made of compatible maneuvers (we can only put a maneuver after another maneuver if the latter ends in a neutral configuration corresponding to the initial configuration of the former one). But the approach remains essentially the same.

References

[1] Barraquand, J and Latombe, J.C., Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles, *Proc. IEEE Int. Conf. Robotics Automation, Sacramento,California,April 1991*,pp.2328-2335,1991.

[2] Campion, G. and d'Andrea-Novel, B. and Bastin, G. , Controllability and state Feedback stabilizability of non holonomic mechanical systems, *Lect. Notes Contr. Inf. Sci., no 162, ed. Canudas de Wit, Advanced Robot Contr., Proc. of the Int. Workshop on nonlin. and adapt. Contr.,Issues in Robotics*, Grenoble/France,Nov. 21-23,1990 pp.106-124.

[3] Campion, G. and d'Andrea-Novel, B. and Bastin, G. , Modelling and state Feedback control of nonholonomic mechanical systems, *Proceedings of tge 30th conference on Decision and Control*, Brighton,England, pp.1184-1189,1990.

[4] Bloch A.M and Reyhanoglu, M and McClamroch, N.,M. , Control and Stabilization of Nonholonomic Dynamic Systems, *IEEE Trans. Automatic Control*,vol 37,no.11, pp.1746-11757, 1992.

[5] Ait-Abderrahim, K. and Samson, C. , Mobile Robot control Part1: Feedback control of an nonholonomic wheeled cart in cartesian space, *INRIA Rapports de Recherche* No. 1288, octobre 1990.

[6] Perl, J., Heuristics - Intelligent Search Strategies for Computer Problem Solving *Addison-Wesley, Reading, Mass.*,1984.

[7] Delebecque, F. and Nikoukhah, R., A mixed symbolic-numeric software environment and its application to control system engineering,*Recent advances in computer aided control*, eds. Hergert, H. and Jamshidi, M., pp. 221-245,1992.

A Appendix

Algorithm

Definition of the variables

Definition of the moving window on the plane on which the path planning and optimization is to be carried out.

Discretization of the window into cells representing a x-y-position and orientation of the vehicle.

Tree and seeds : Branches of the tree are maneuvers. The end of each tree may define the beginning of a new tree. The beginning of a new tree will be called "seed". Each seed is represented by:

i. its position and orientation on the plane
ii. its parent seed
ii. maneuvers (tree branches) leading from the parent seed to this seed.
iii. accumulated cost

The paht-tree is stored in a lists associated to the cells. If for the first time a subpath reaches a cell, the cell is associated a list containing the following items:

i. position of a not developed seed in this cell
ii. list of the positions of former already developed seeds of the cell
iii. accumulated cost to reach this cell

If a subpath reaches a cell which is already associated a list, its list is updated.

"openlist" ordered list of the addresses of cells containing a seed to be developed. The order corresponds to the value of accumulated cost.

"depth" defines the number of branching in the tree.

Initialization

Definition of the starting point:

Initialization of the grid: To none of the cells is associated a list.

Definition of the starting point of tree as a seed. Initialization of the corresponding cell (association of a list to the corresponding cell containing the first seed of the path-tree).

Initialization of "openlist": openlist = [address of cell with the first seed].

Controller

main programm

While the goal position is not reached *do*

While openlist $\neq$ nil *do*

seed = first element of openlist

Remove seed from openlist.

[listofnewseeds]=**treebranching**[seed,depth]

Declare seed in corresponding cell as developed and add it to list of former seeds in corresponding cell.

[openlist,finalcell]= **seedplacing**[openlist,listofnewseeds]

if finalcell = nil *then* reinitialize openlist and continue with a higher value for depth (the tree will be richer then in the previous step).

end

[listofmaneuvers]=**backtrace**[finalcell]

Apply the first n maneuvers of listofmaneuvers to the system.

Shift the window to the new position of the system.

Reinitialization of the grid.

Update information ot the environment.

New position of the vehicle defines the new starting point.

openlist = [number of cell of the new firstseed].

end

[listofneeseeds]=**treebranching**[seed,depth]

listofseeds = nil

if cost of seed < cost in cell *and* (if any) cost of finalcell *then*

develop seed into a tree.

For concatenation of maneuvers defined by the seed *do*

if the subpath leading from the seed to the end of the concatenation of maneuvers does not pass by or ends in the obstacles *then* add its end as new seed to listofseeds

end

[openlist,finalcell]=**seedplacing**[openlist,listofnewseeds]

For all elements of listofnewseeds *do*

if the cost of the seed is < the actual cost in the cell *or* the cell is still empty *then*

if the seed is in the goal-position *and* its cost < then the actual cost of finalcell *then* replace the seed in finalcell by the new seed.

elseif the seed is in goal-position but with a cost > the actual final cost *then* throw away the seed.

else Place the seed in the corresponding cell as candidate for a new branching in grid and add the cell number to openlist.

The elements of openlist are order such that the seed with least cost is at first position.

[listofmaneuves]**backtracing**[finalcell]

While the firstseed is not a member of the list of developed seeds of the actual cell *do*

Take parent seed of the actual seed out of list of developed seeds of the actual cell.

Set actual seed = parent seed and take corresponding cell as actual cell.

end

B Considered model of a bicycle

Given the Lagrangien of the whole multibody system as a sum of the Lagrangiens of each independent body and a set of holonomic and nonholonoimc constraints describing the contacts between the bodies and the contacts vehicle-ground, we obtain the following implicit dynamical model of the bicycle.

$$\frac{\partial L(\mathbf{q},\dot{\mathbf{q}})}{\partial \mathbf{q}} - \frac{d}{dt}\frac{\partial L(\mathbf{q},\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} + \left(\frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}}\right)^T \mathbf{\Lambda} + \mathbf{G}^T(\mathbf{q})\mathbf{\Gamma} = 0 \ ,$$

$$\mathbf{f}(\mathbf{q}) = 0 \ ,$$

$$\mathbf{G}(\mathbf{q})\dot{\mathbf{q}} = 0 \ ,$$

which results with $\mathbf{k} = \mathbf{K}(\mathbf{q})\mathbf{u}$ as the vector of external and internal forces and torques (controls) in the more condensed form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{k} + \mathbf{f}_q^T(\mathbf{q})\mathbf{\Lambda} + \mathbf{G}^T\mathbf{\Gamma}$$

$$\mathbf{f}(\mathbf{q}) = 0 \quad ,$$

$$\mathbf{G}(\mathbf{q})\dot{\mathbf{q}} = 0 \quad .$$

For the modelization we have used the following set of 23 variables (see figure 4.1):

-center of gravity of rear wheel:
x_1 , y_1 , z_1 (coordinates of P_1)
-Euler angles of rear wheel and frame:
ϕ_1 (direction of the wheel)
θ_1 (lean, the up position is $\frac{\pi}{2}$)
ψ_1 (turning of the wheel)
-center of gravity of frame:
x_2 , y_2 , z_2 (coordinates of P_2)
-angle between horizontal and frame:
ξ_2
-fork:P_3
x_3 , y_3 , z_3 (coordinates of P_3)
-angle between horizontal and fork - front projection:
ξ_3
-front projection :
x_4 , y_4 , z_4 (coordinates of P_4)
-Euler angles fork, front projection and front wheel:
ϕ_4 , θ_4 ψ_4
-center of gravity of front wheel:
x_5 , y_5 , z_5 (coordinates of P_5) .

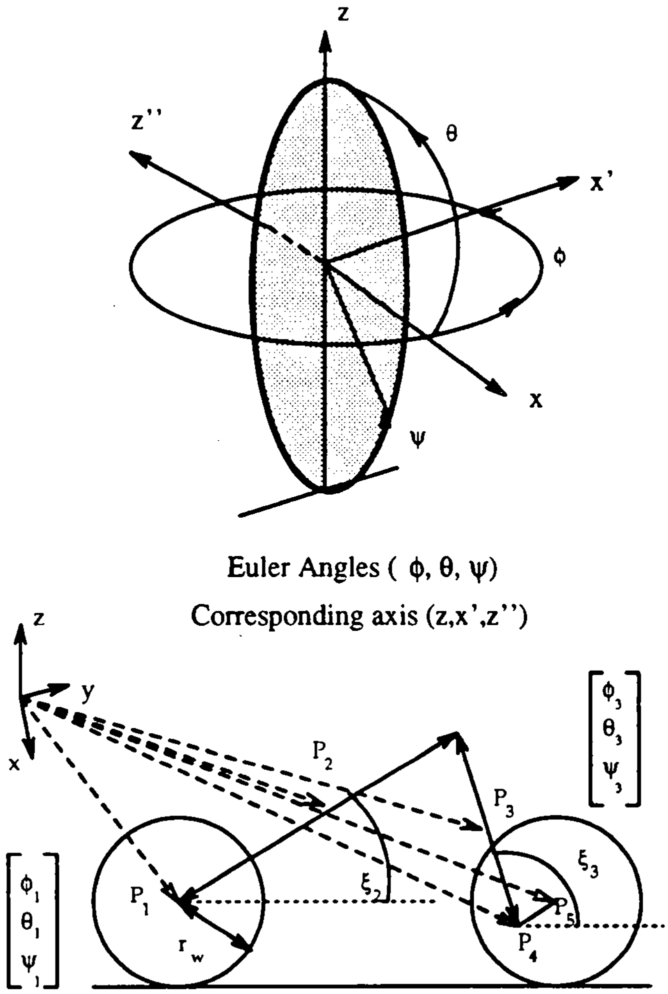


Figure B.1 *Parametrization of the model.*

$$
\begin{aligned}
l_1 &= P_1 P_4 & l_6 &= P_4 P_5 \\
l_2 &= P_1 P_2 & l_4 &= 2 l_1 \\
l_3 &= P_4 P_3 & l_6 &= 2 l_3
\end{aligned}
$$

Within the used set of variables the Lagrangian function is :

$$
\begin{aligned}
L \;=\; & \tfrac{1}{2}\Big(I_{rwn}\left(\dot\theta_1^2+\dot\phi_1^2\sin(\theta_1)^2\right)+I_{rwr}\left(\dot\phi_1\cos(\theta_1)+\dot\psi_1\right)^2+ \\
& +m_{rw}\,(\dot x_1^2+\dot y_1^2+\dot z_1^2)-\;2\;\;m_{rw}gr_w\sin(\theta_1)+ \\
& +I_{frn}\left(\left(-\dot\theta_1\sin(\xi_2)+\dot\phi_1\sin(\theta_1)\cos(\xi_2)\right)^2+\left(\dot\phi_1\cos(\theta_1)+\dot\xi_2\right)^2\right)+ \\
& +I_{frr}\left(\dot\theta_1\cos(\xi_2)+\dot\phi_1\sin(\theta_1)\sin(\xi_2)\right)^2+ \\
& m_{fr}\,(\dot x_2^2+\dot y_2^2+\dot z_2^2)-2\;\;m_{fr}g\,(r_w+l_2\sin(\xi_2))\sin(\theta_1)+ \\
& +I_{rwn}\left(\left(-\dot\theta_4\sin(\xi_3)+\dot\phi_4\sin(\theta_4)\cos(\xi_3)\right)^2+\left(\dot\phi_4\cos(\theta_4)+\dot\xi_3\right)^2\right)+ \\
& +I_{forr}\left(\dot\theta_4\cos(\xi_3)+\dot\phi_4\sin(\theta_4)\sin(\xi_3)\right)^2+ \\
& +m_{fo}\,(\dot x_3^2+\dot y_3^2+\dot z_3^2)-2\;\;m_{fo}g\,(r_w+l_3\sin(\xi_3))\sin(\theta_4)+ \\
& +I_{fwn}\left(\dot\theta_4^2+\dot\phi_4^2\sin(\theta_4)^2\right)+I_{fwr}\left(\dot\phi_4\cos(\theta_4)+\dot\psi_4\right)^2+ \\
& +m_{fw}\,(\dot x_5^2+\dot y_5^2+\dot z_5^2)-2\;\;m_{fw}gr_w\sin(\theta_4)\Big)
\end{aligned}
$$

The dynamic model has to verify the constraints $\mathbf{G}(\mathbf{q})\dot{\mathbf{q}} = 0$ and $\mathbf{f} = 0$:
Nonholonomic constraints: they result from the condition of roling without sliding for the wheels.

$$\Delta \mathbf{v} = \mathbf{v}_j + \omega_i \times \mathbf{r}_w = 0 \tag{B.27}$$

with $\omega_{\rm i}$ $(i = 1)$ as angular velocity of the rear wheel v_j $(j = 1)$, the velocity of P_1 and $\mathbf{r}_w$ the radius-vector of the rear wheel. Indicating by the matrix A^z_ϕ a rotation of the angle ϕ around the z-axis (or the x-axis after a former rotation around the z-axis, we than write x') from the fixed system of coordinates to the rotated (and mouving) one we find the following local representations for ω_i and $\mathbf{r}_w$:

$$\omega_i = \begin{pmatrix} 0 \\ 0 \\ \dot{\phi}_i \end{pmatrix} + (A^z_{\phi_i})^T \left[\begin{pmatrix} \dot{\theta}_i \\ 0 \\ 0 \end{pmatrix} + (A^{x'}_{\theta_i})^T \begin{pmatrix} 0 \\ 0 \\ \dot{\psi}_i \end{pmatrix} \right] = \begin{pmatrix} \dot{\theta}_i \cos(\phi_i) + \dot{\psi}_i \sin\theta_i \sin(\phi_i) \\ \dot{\theta}_i \sin(\phi_i) - \dot{\psi}_i \sin(\theta_i)\cos(\phi_i) \\ \dot{\phi}_i + \dot{\psi}_i \cos(\theta_i) \end{pmatrix},$$

$$\mathbf{r}_w = (A^z_{\phi_i})^T (A^{x'}_{\theta_i})^T \begin{pmatrix} 0 \\ -r_w \\ 0 \end{pmatrix} = \begin{pmatrix} r_w \cos(\theta_i)\sin(\phi_i) \\ -r_w \cos(\theta_i)\cos(\phi_i) \\ -r_w \sin(\theta_i) \end{pmatrix},$$

with $\mathbf{A}^z_\phi$ and $\mathbf{A}^{x'}_\theta$ as

$$\mathbf{A}^z_{\phi_i} = \begin{pmatrix} \cos(\phi_i) & \sin(\phi_i) & 0 \\ -\sin(\phi_i) & \cos(\phi_i) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{A}^{x'}_{\theta_i} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_i) & \sin(\theta_i) \\ 0 & -\sin(\theta_i) & \cos(\theta_i) \end{pmatrix}.$$

Condition (B.27) results for the rear wheel ($i = 1$ and $j = 1$) in two nonholonomic and one holonomic constraints. If we chose for the front wheel the same radius as for the rear wheel, we obtain with $i = 4$ and $j = 5$ two further nonholonomic and one holonomic constrains. The nonholnomic constraints are collected in $\mathbf{G}(\mathbf{q})\dot{\mathbf{q}} = 0$ for two holonomic constraints see the first two components of $\mathbf{f}(\mathbf{q})$).

$$\mathbf{G}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \dot{x}_1 + r_w \left(\dot{\phi}_1 \cos(\theta_1)\cos(\phi_1) - \dot{\theta}_1 \sin(\phi_1)\sin(\theta_1) + \dot{\psi}_1 \cos(\phi_1) \right) \\ \dot{y}_1 + r_w \left(\dot{\phi}_1 \cos(\theta_1)\sin(\phi_1) + \dot{\theta}_1 \cos(\phi_1)\sin(\theta_1) + \dot{\psi}_1 \sin(\phi_1) \right) \\ \dot{y}_5 + r_w \left(\dot{\phi}_4 \cos(\theta_4)\sin(\phi_4) + \dot{\theta}_4 \cos(\phi_4)\sin(\theta_4) + \dot{\psi}_4 \sin(\phi_4) \right) \\ \dot{x}_5 + r_w \left(\dot{\phi}_4 \cos(\theta_4)\cos(\phi_4) - \dot{\theta}_4 \sin(\phi_4)\sin(\theta_4) + \dot{\psi}_4 \cos(\phi_4) \right) \end{bmatrix}$$

Holonomic constraints: equations (3)-(5), (6)-(8) and (14)-(16) describe the links between respectively P_1P_2, P_4P_3 and P_4P_5. We have just three equations for each joint as the Euler angles are simular for P_1 and P_2 (P_3, P_4 and P_5). The remaining five equations (9)-(13) link the rear-part with the front-part, i.e. P_1P_4. There are five equations as the joint has one degree of freedom.

$$
\mathbf{f}(\mathbf{q}) = \begin{bmatrix}
z_1 - r_w \sin(\theta_1) & (1) \\
z_5 - r_w \sin(\theta_4) & (2) \\
x_1 - x_2 + l_2 (\cos(\phi_1)\cos(\xi_2) - \sin(\phi_1)\sin(\xi_2)\cos(\theta_1)) & (3) \\
y_1 - y_2 + l_2 (\sin(\phi_1)\cos(\xi_2) + \cos(\phi_1)\sin(\xi_2)\cos(\theta_1)) & (4) \\
z_1 - z_2 + l_2 \sin(\xi_2)\sin(\theta_1) & (5) \\
z_4 - z_3 + l_3 \sin(\xi_3)\sin(\theta_4) & (6) \\
y_4 - y_3 + l_3 (\sin(\phi_4)\cos(\xi_3) + \cos(\phi_4)\sin(\xi_3)\cos(\theta_4)) & (7) \\
x_4 - x_3 + l_3 (\cos(\phi_4)\cos(\xi_3) - \sin(\phi_4)\sin(\xi_3)\cos(\theta_4)) & (8) \\
\begin{array}{l} l_4 (\cos(\phi_1)\cos(\xi_2) - \sin(\phi_1)\sin(\xi_2)\cos(\theta_1)) - \\ \quad - l_5 (\cos(\phi_4)\cos(\xi_3) - \sin(\phi_4)\sin(\xi_3)\cos(\theta_4)) - x_4 + x_1 \end{array} & (9) \\
\begin{array}{l} l_4 (\sin(\phi_1)\cos(\xi_2) + \cos(\phi_1)\sin(\xi_2)\cos(\theta_1)) - \\ \quad - l_5 (\sin(\phi_4)\cos(\xi_3) + \cos(\phi_4)\sin(\xi_3)\cos(\theta_4)) - y_4 + y_1 \end{array} & (10) \\
l_4 \sin(\xi_2)\sin(\theta_1) - l_5 \sin(\xi_3)\sin(\theta_4) - z_4 + z_1 & (11) \\
(x_1 - x_4)^2 + (y_1 - y_4)^2 + (z_1 - z_4)^2 - l_1^2 & (12) \\
\begin{array}{l} l_5(\sin(\theta_1)\sin(\phi_1 - \phi_4)\cos(\xi_3) - \sin(\theta_1 - \theta_4)\sin(\phi_1 - \phi_4)\sin(\xi_3) + \\ \quad + \cos(\theta_1)\sin(\xi_3)\sin(\theta_4)) \end{array} & (13) \\
x_5 - x_4 - l_6 (\cos(\phi_4)\sin(\xi_3) - \cos(\theta_4)\sin(\phi_4)\cos(\xi_3)) & (14) \\
y_5 - y_4 - l_6 (\sin(\phi_4)\sin(\xi_3) + \cos(\theta_4)\cos(\phi_4)\cos(\xi_3)) & (15) \\
z_5 - z_4 + l_6 \cos(\xi_3)\sin(\theta_4) & (16)
\end{bmatrix}
$$

For the vector of external and internal forces and torques we obtain the following vector (u_1 is the torque on the handlbars, u_2 a torque on the axle of the rear wheel to accelerate the bicycle):

$$
\mathbf{k} = c \begin{bmatrix} 0 \\ 0 \\ 0 \\ -u_1 \sin(\theta_1)\cos(\phi_1 - \phi_4) \\ -u_1 \cos(\theta_1)\sin(\phi_1 - \phi_4) \\ u_2 \\ -u_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_1 \cot(\xi_3)\sin(\theta_1)\sin(\phi_1 - \phi_4) \\ 0 \\ 0 \\ 0 \\ u_1 \sin(\theta_1)\cos(\phi_1 - \phi_4) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

$$
c = \frac{1}{\sin(\theta_4)\sin(\theta_1)\cos(\phi_1 - \phi_4) + \cos(\theta_4)\cos(\theta_1)}
$$

The components 4, 5, 6, 7, 14 and 19 act on ϕ_1, θ_1, ψ_1, ξ_2, ξ_3 and θ_4 respectively.

Unité de Recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399

RR.1955